Fábrica de Noobs

Desmistificando: loop temporal de 1969

Natanael Antonioli

Outubro de 2025

1. SUMÁRIO

2. Com o que estamos lidando? 3

3. O que há de especial em 1969? 4

4. Mas por que esse valor aparece de forma recorrente? 6

5. Mas por que uma criadora de conteúdo teve a transmissão banida? 8

6. 1969 tem o mesmo calendário que 2025? Isso é especial? 9

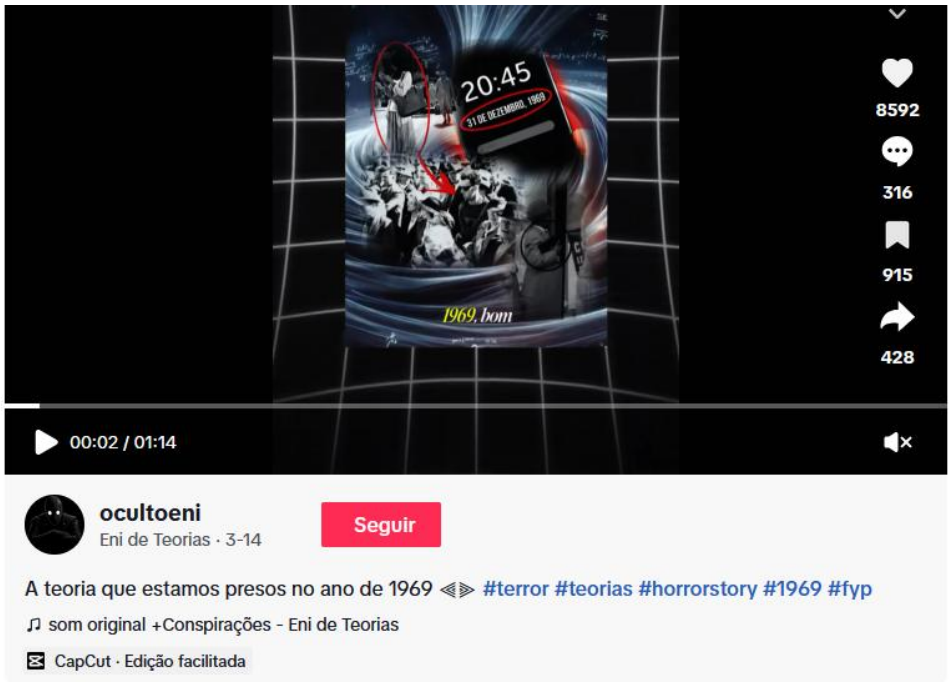
7. Conclusões 10

2. COM O QUE ESTAMOS LIDANDO?

Uma pessoa, no TikTok, foi banida por publicar uma live em 1969, quando a internet sequer existia! Isso poderia ser um erro pontual, mas pessoas observam o ano de 1969 em ligações telefônicas, mensagens de texto e postagens em múltiplos aplicativos. Como 1969 foi marcado por eventos importantes e o calendário de 1969 é idêntico ao de 2025, suspeita-se que vivemos em um loop temporal iniciando-se em 1969.

A reprodução da história fica por conta de um vídeo no TikTok (https://www.tiktok.com/@ocultoeni/video/7481682317388336439?_t=ZM-90qt6yu0nQe&_r=1).

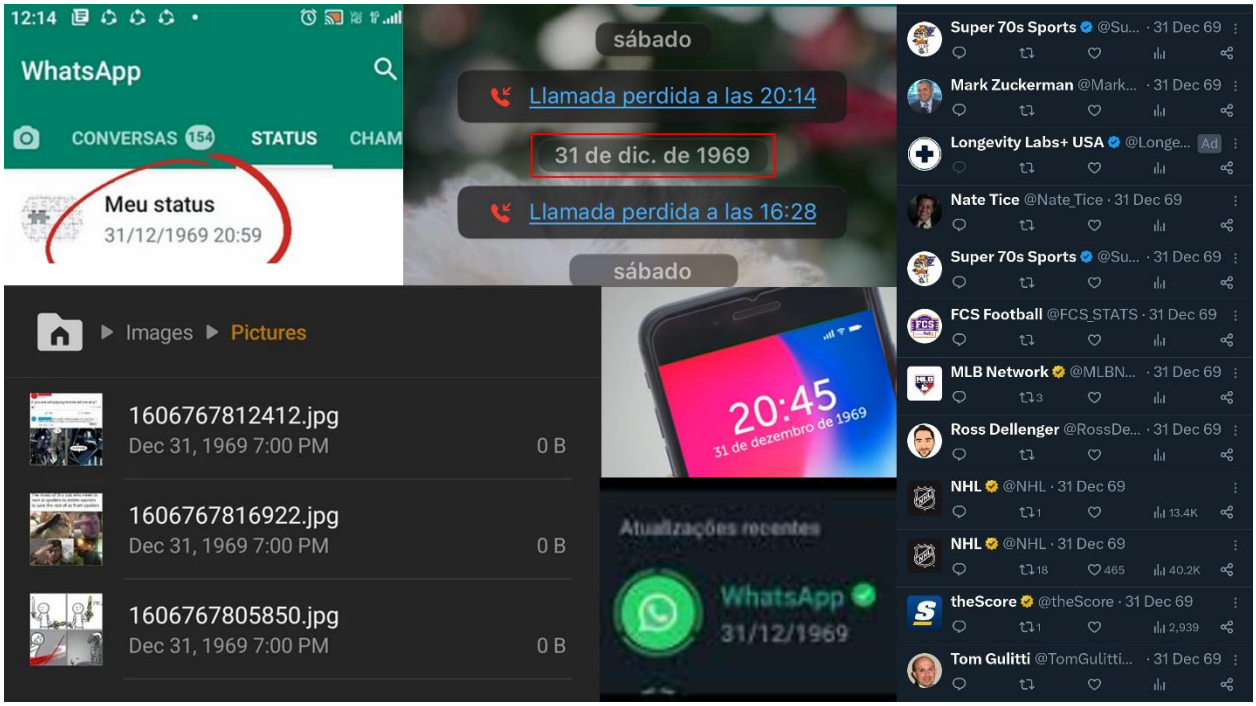
Figura 1: captura do vídeo.



3. O QUE HÁ DE ESPECIAL EM 1969?

Há ocorrências demais de sistemas exibindo o ano de 1969 – mais especificamente 31 de dezembro de 1969 – para considerarmos que esse comportamento é meramente um erro de data com valor aleatório. Ainda que **um loop temporal seja uma hipótese bastante ousada**, é seguro concluir que **há algo especial sobre essa data**.

Figura 2: múltiplas instâncias do mesmo erro.



Um dos principais aspectos da computação é o armazenamento de dados. Alguns desses dados, como data e hora, **representam desafios adicionais** por diversas razões:

I. São **compostos de seis valores**, sendo eles ano, mês, dia, hora, minuto e segundo;

II. Possem **operações aritméticas não triviais**, uma vez que devem levar em consideração meses com 28, 29, 30 ou 31 dias;

III. Possuem **múltiplas representações graças aos fuso-horários**.

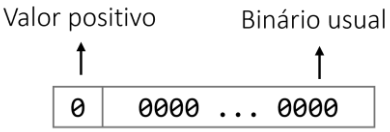
A data Unix armazena **apenas um valor inteiro em segundos** que representa **o total de segundos depois das 00:00 do dia 1 de janeiro de 1970 no fuso-horário UTC**, em que cada dia transcorrido representa 86400 segundos.

Dessa forma, um sistema que utilize a data Unix apenas precisa armazenar um valor de 64 bits, uma arquitetura **tão simples que é universal**, para a data. Além, disso, quaisquer operações aritméticas **são feitas utilizando os valores em segundos**, e a conversão para uma data legível por seres humanos **é feita utilizando regras e calendários armazenados**.

A forma como o valor é armazenado **é chamada complemento de dois**. Nessa forma, usamos uma notação diferente para números positivos e negativos. Em números positivos, o bit mais significativo será 0, e os 63 bits seguintes serão a representação binária do número; ao passo que em números negativos, o bit mais significativo será 1, e os 63 bits seguintes **serão a representação binária do número invertida, e então somada de 1**.

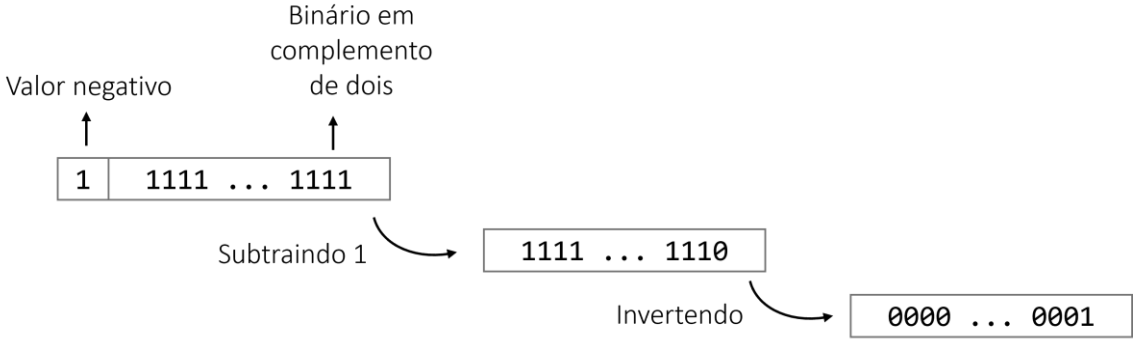
Quando todos os bits armazenados são zero, **a data em segundos é 0** e representa exatamente a data base, ou seja, **1 de janeiro de 1970, às 00:00**.

Figura 3: caso em que todos os bits são 0.



Quando todos os bits armazenados são um, **a data em segundos é -1** e representa exatamente 1 segundo antes da data base, ou seja, **31 de dezembro 1969, às 23:59**.

Figura 4: caso em que todos os bits são 1.



4. MAS POR QUE ESSE VALOR APARECE DE FORMA RECORRENTE?

A data Unix **foi desenvolvida em 1971 para sistemas Unix** e foi utilizada nas bibliotecas de tempo **da linguagem C e da API Unix**. A partir de 1983, o projeto GNU reimplementou uma série de funções Unix e **manteve o mesmo padrão de data**.

Em 1991, Linus Torvalds escreveu o núcleo Linux, que foi então combinado com as funções GNU. Por essa razão, sistemas operacionais atuais com núcleo Linux – como Debian, Ubuntu, Kali, Mint, Red Hat, Arch e Gentoo – **usam esse mesmo padrão** e rodam não só em dispositivos pessoais, **mas também em servidores**.

Já o macOS **também usa o mesmo padrão** porque foi construído a partir de uma sucessão de sistemas que remontam ao BSD, desenvolvido a partir do código fonte do Unix. Android, por sua vez, **é simplesmente um núcleo Linux modificado**, e **herda o padrão**.

A linguagem C++ é um superconjunto da linguagem C, então **também usa essa função**. Como o interpretador de Python é escrito em C, **a mesma função é naturalmente usada**. Aplicativos escritos em Java deveriam rodar em múltiplos sistemas e, por isso, a Máquina Virtual Java **utiliza as bibliotecas C/C++**, e **usa o mesmo padrão**. Como múltiplos módulos foram feitos em Java, **Javascript adotou o mesmo padrão**.

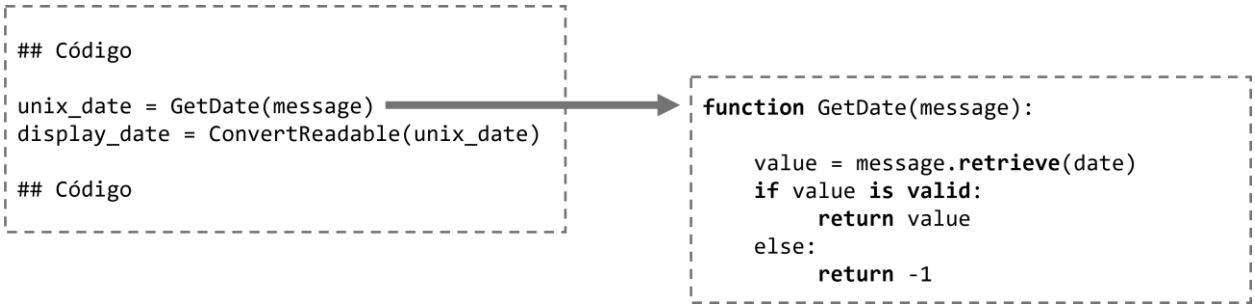
Por fim, Windows não utiliza esse padrão (e sim uma época cujo 0 mapeia para às 00:00 de 1 de janeiro de 1601), mas **possui uma função de conversão que devolve data Unix para aplicações** o que significa que, mesmo em um computador pessoal com Windows, **a data Unix ainda se faz presente**.

Assim, dada a imensa quantidade de sistemas que usam esse padrão, **é muito provável que o erro seja comum** – e não restrito a uma mesma aplicação. Mas por que especificamente -1, e não outros valores – ao menos não com tamanha frequência?

Um sistema com algum grau de complexidade **provavelmente usará uma função para obter a data de um objeto** (como uma postagem, mensagem ou arquivo). Essa função é chamada pelo programa, e então **devolve a ele um resultado**, que pode ser um único número – nesse caso, a data Unix. Então, outra função **transforma o valor numérico em uma data que humanos podem ler com facilidade**.

Porém, **esse tipo de função pode falhar**. Nesse caso, para evitar que um valor imprevisível seja devolvido, **é comum que funções retornem um valor indicativo de erro**. Muitos desenvolvedores escolhem esse valor como -1.

Figura 5: pseudocódigo ilustrando esse funcionamento.



Assim, quando a função que recupera a data falha, um valor -1 é devolvido para a função que a converte para data humanamente legível. E, nesse caso, **a data humanamente legível a partir de -1 é 23:59 de 31 de dezembro de 1969**.

Logo, esses erros ocorrem quando **há falha de comunicação entre os processos em alguma etapa antes da exibição para o usuário**. Em algumas situações, há uma conversão de fuso-horário, o que **pode adicionar ou subtrair horas do resultado**, inclusive **avançando para 1970**.

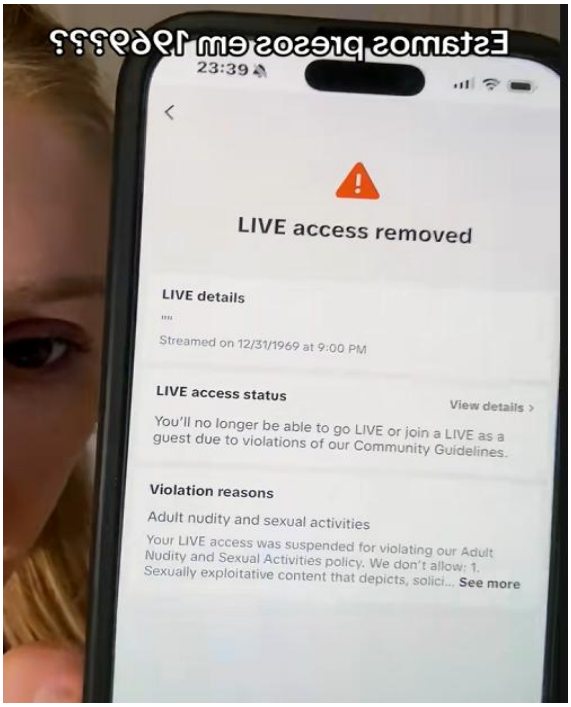
5. MAS POR QUE UMA CRIADORA DE CONTEÚDO TEVE A TRANSMISSÃO BANIDA?

Uma das motivações para a teoria é que algumas pessoas tiveram transmissões ao vivo no TikTok banidas. Uma dessas pessoas foi a criadora de conteúdo Cela Lopes (https://www.tiktok.com/@celaalopes/video/7470728302815087878).

Ela afirma que a data foi 31 de dezembro de 1969 às 9 da manhã, mas a imagem do celular mostra que **a data é, na verdade, 9 da noite** – o *PM* indica horário após o meio-dia. Esse horário é 3 horas antes da meia noite e, para converter horários UTC para Brasília **deve-se subtrair 3 horas**, o que indica que a leitura original é de -1, seguida por um ajuste de fuso-horário.

Quanto ao motivo do banimento, **a mensagem informa nudez ou atividades sexuais**. Ao passo que redes sociais são conhecidas por banimentos arbitrários – e não sabemos se esse foi o caso – **o erro de exibição da data não possui nenhuma relação com a medida**.

Figura 6: tela do celular. Invertida para visualização.


Estamos presos em 1969???
23:39
LIVE access removed
LIVE details
Streamed on 12/31/1969 at 9:00 PM
LIVE access status
View details
You'll no longer be able to go LIVE or join a LIVE as a guest due to violations of our Community Guidelines.
Violation reasons
Adult nudity and sexual activities
Your LIVE access was suspended for violating our Adult Nudity and Sexual Activities policy. We don't allow: 1. Sexually exploitative content that depicts, solici... See more

6. 1969 TEM O MESMO CALENDÁRIO QUE 2025? ISSO É ESPECIAL?

Tentativas de associação do ano de 1969 com 2025 sequer possuem sentido lógico, **uma vez que o erro ocorre há anos**. Porém, ao passo que o calendário de 2025 realmente é o mesmo que o de 1969 em termos da paridade dia do mês e dia da semana, esse **é um fenômeno comum** que ocorre combinando-se o total de dias do ano em anos usuais e bissextos.

Em https://www.timeanddate.com/calendar/repeating.html?year=2025, pode-se consultar repetições. Verificaremos que o calendário de 2025 **é igual ao de 1941, 1947, 1958, 1969, 1975, 1986, 1997, 2003, 2014** e, nos anos seguintes, **de 2031, 2042 e 2036**.

Figura 7: repetições de calendário.

1941	84 years before	+ 11 years
1947	78 years before	+ 6 years
1958	67 years before	+ 11 years
1969	56 years before	+ 11 years
1975	50 years before	+ 6 years
1986	39 years before	+ 11 years
1997	28 years before	+ 11 years
2003	22 years before	+ 6 years
2014	11 years before	+ 11 years
2025	selected year	+ 11 years
2031	6 years after	+ 6 years
2042	17 years after	+ 11 years
2053	28 years after	+ 11 years
2059	34 years after	+ 6 years
2070	45 years after	+ 11 years
2081	56 years after	+ 11 years
2087	62 years after	+ 6 years

7. CONCLUSÕES

Assim, podemos concluir, em primeiro lugar, que imaginar um loop temporal ao se deparar com um bug de computador **não parece uma atitude cientificamente razoável**. Além disso, tal bug pode ser explicado facilmente por **uma falha de comunicação no aplicativo combinada com a forma pela qual a maioria dos sistemas do mundo lida com datas**.